

Machine Learning Operations Semester Project-Task 1

Environmental Monitoring and Pollution Prediction System



Student Name: Moeed Asif

Roll No: 21i-0483

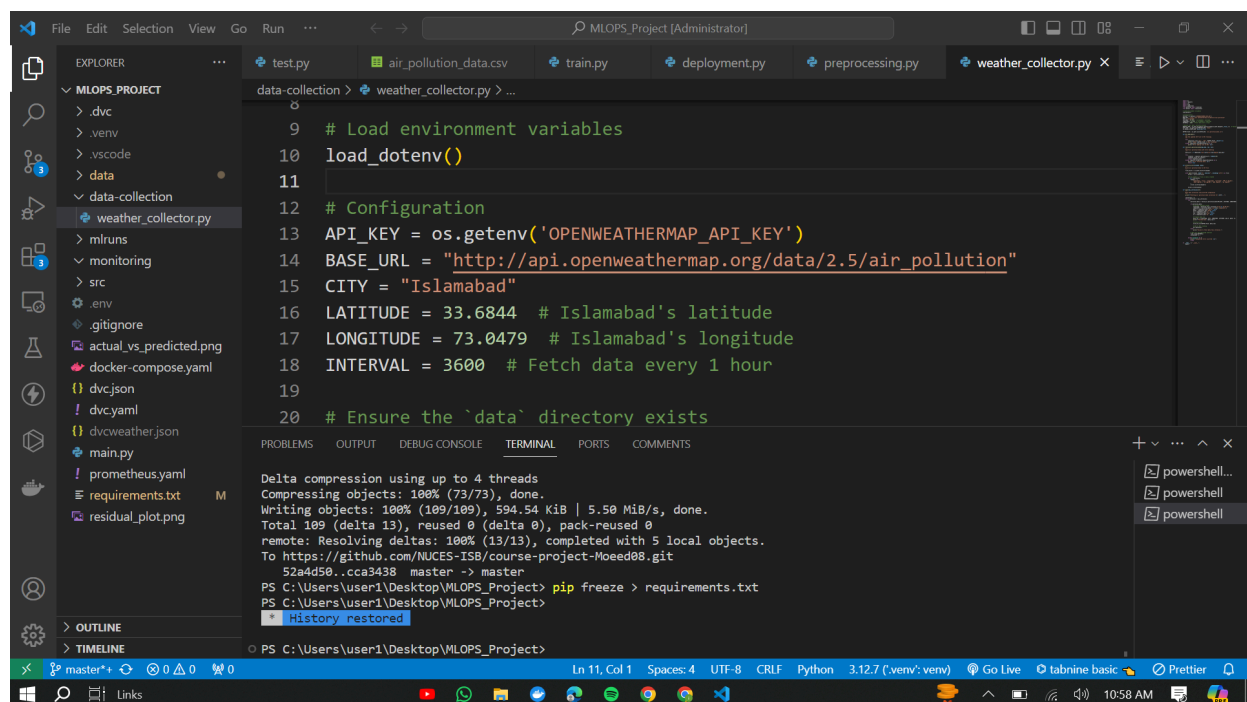
Section : Mlops-B

Introduction:

Data version control (DVC) plays a crucial role in the maintenance of data for the training of machine learning models. DVC uses version control techniques in collaboration with the github to manage the versions of the data. In this task we used DVC and Github to do version controlling of the system(Weather data collection and version controlling).

Research Live Streams Data:

I have searched and tried many different websites for the live weather data collection but ended up with the **OpenWeatherMap API**. The main benefit of this API is that it can easily integrate with the python libraries, and the website provides upto 500+ free of the API calls for data collection, just simply create the API and use it in your code. I have placed the API in my `.env` file and loaded it in my data collection script.



```
9 # Load environment variables
10 load_dotenv()
11
12 # Configuration
13 API_KEY = os.getenv('OPENWEATHERMAP_API_KEY')
14 BASE_URL = "http://api.openweathermap.org/data/2.5/air_pollution"
15 CITY = "Islamabad"
16 LATITUDE = 33.6844 # Islamabad's latitude
17 LONGITUDE = 73.0479 # Islamabad's longitude
18 INTERVAL = 3600 # Fetch data every 1 hour
19
20 # Ensure the `data` directory exists
```

```
Delta compression using up to 4 threads
Compressing objects: 100% (73/73), done.
Writing objects: 100% (109/109), 594.54 KiB | 5.50 MiB/s, done.
Total 109 (delta 13), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (13/13), completed with 5 local objects.
To https://github.com/NUCES-ISB/course-project-Moeeed08.git
52a4d50..cca3438 master -> master
PS C:\Users\User1\Desktop\WLOPS_Project> pip freeze > requirements.txt
PS C:\Users\User1\Desktop\WLOPS_Project>
History restored
```

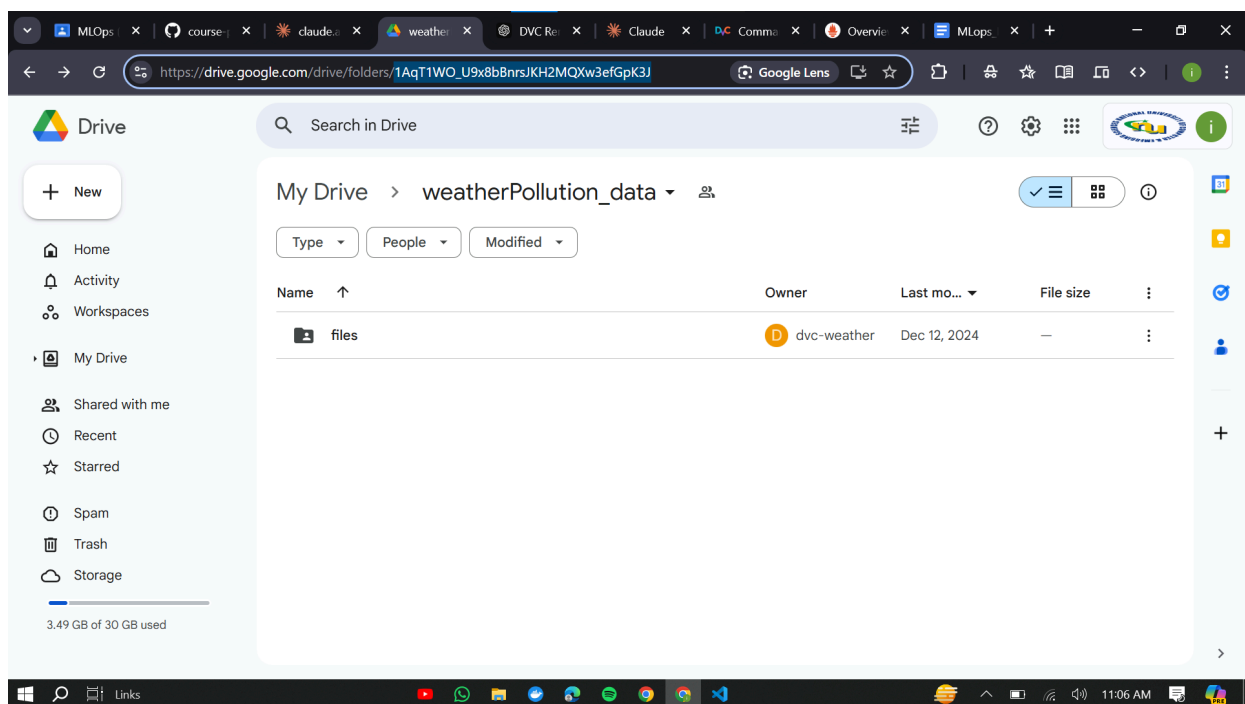
DVC Repository Setup:

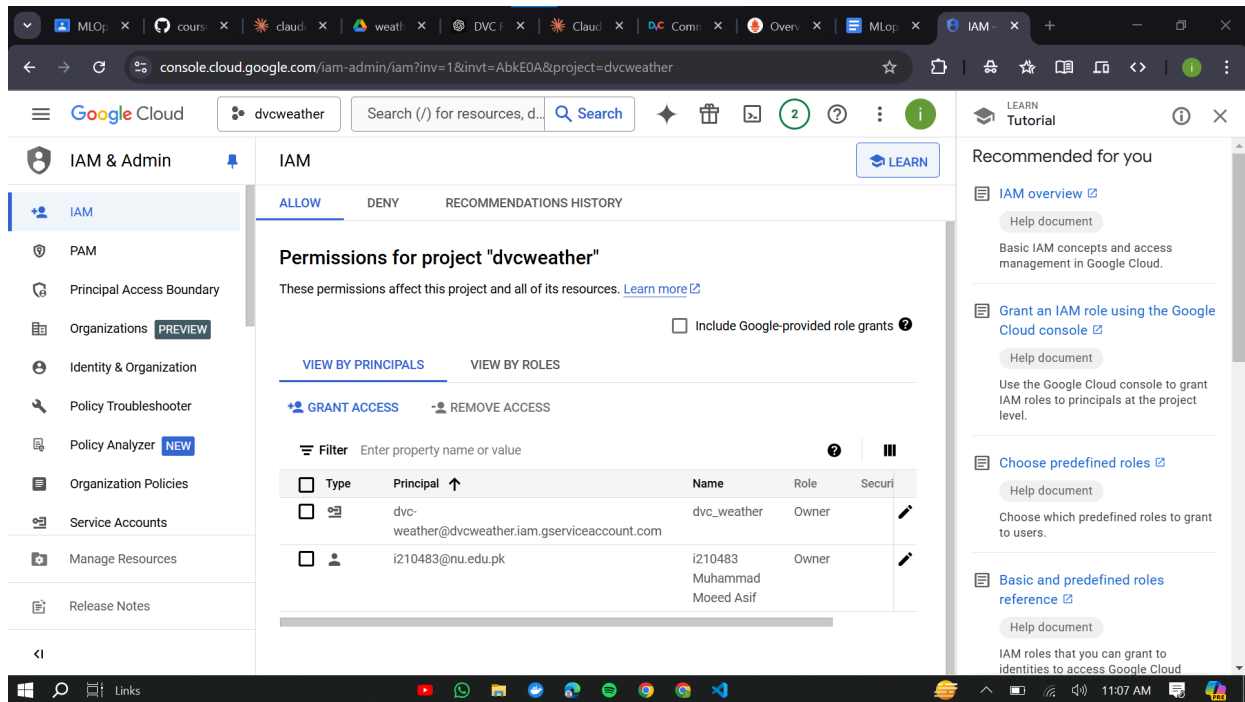
Make a virtual environment in the python and run the command **pip install dvc**, this will set up the data version control system in the project. After installing dvc, make an empty github

repository in your local machine by running **git init**. Next run the command **dvc init** and it will create a repository for the data version control.

Remote Storage configuration:

Make a remote storage configuration using dvc with google drive. For this purpose, we first install **dvc-[gdrive]** into the system to efficiently interact with the google drive, next in the google cloud console, we make a service account and enabled the **API**. Make an empty folder in the google drive and copy the folder id from the URL like in the screenshots below.





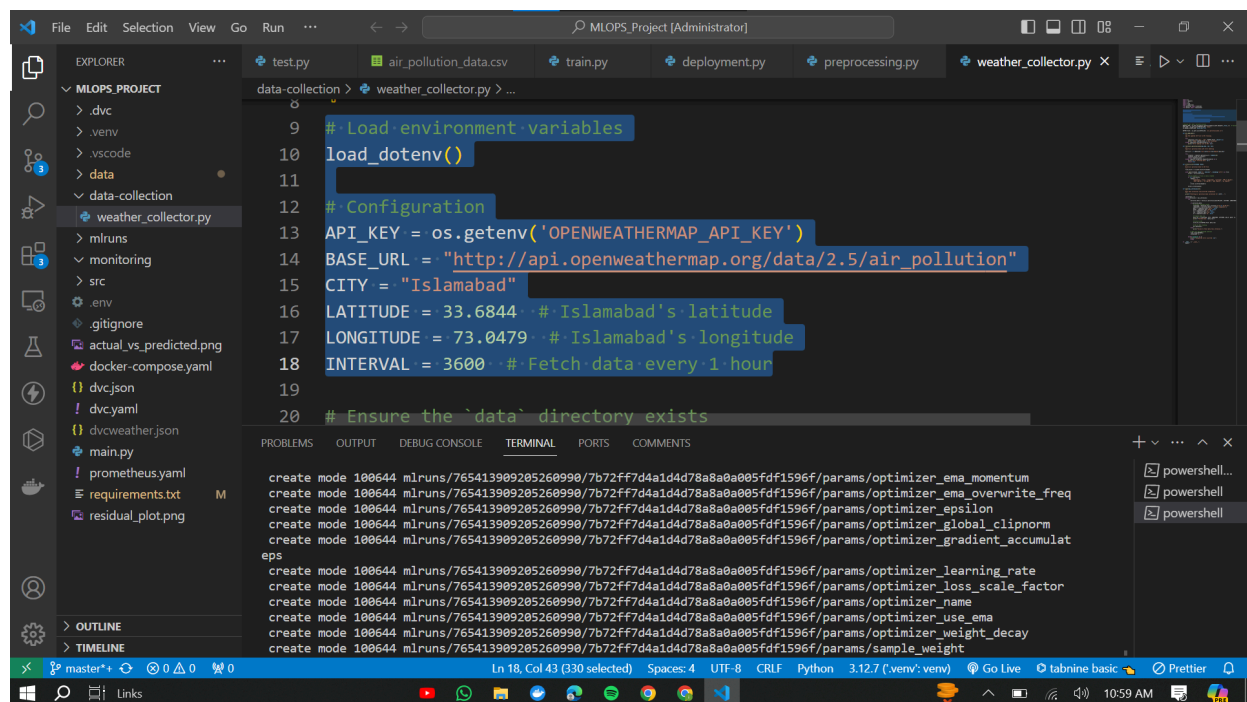
Next we make an empty dvc repository with **dvc init** and run the following commands to connect the dvc with my google drive for remote storage

1. **dvc remote add -d myremote gdrive://1AqT1WO_U9x8bBnrsJKH2MQXw3efGpK3J**
2. **dvc remote modify myremote gdrive_use_service_account true**
3. **dvc remote modify dvc_remote gdrive_service_account_json_file_path C:\Users\user1\Desktop\MLOPS_Project\dvc.json**

Data collection Script:

Make a python script which loads the api key given by the [OpenWeathermap](https://openweathermap.org/api) and setup the environment in python for the data collection such as **base url** is "<http://api.openweathermap.org/data/2.5/weather>",

City= “Islamabad”, LATITUDE, LONGITUDE”, and Interval= 3600 which fetch the data every hour.

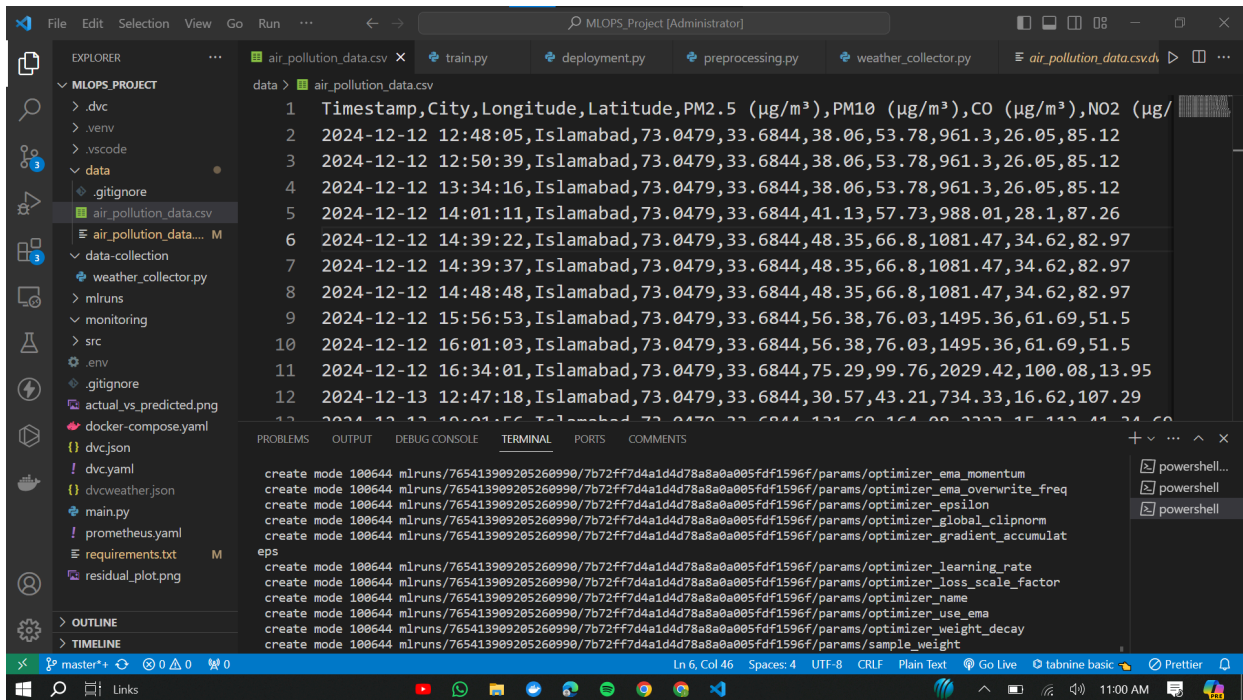


```
9  # Load environment variables
10 load_dotenv()
11
12 # Configuration
13 API_KEY = os.getenv('OPENWEATHERMAP_API_KEY')
14 BASE_URL = "http://api.openweathermap.org/data/2.5/air_pollution"
15 CITY = "Islamabad"
16 LATITUDE = 33.6844 # Islamabad's latitude
17 LONGITUDE = 73.0479 # Islamabad's longitude
18 INTERVAL = 3600 # Fetch data every 1 hour
19
20 # Ensure the `data` directory exists
```

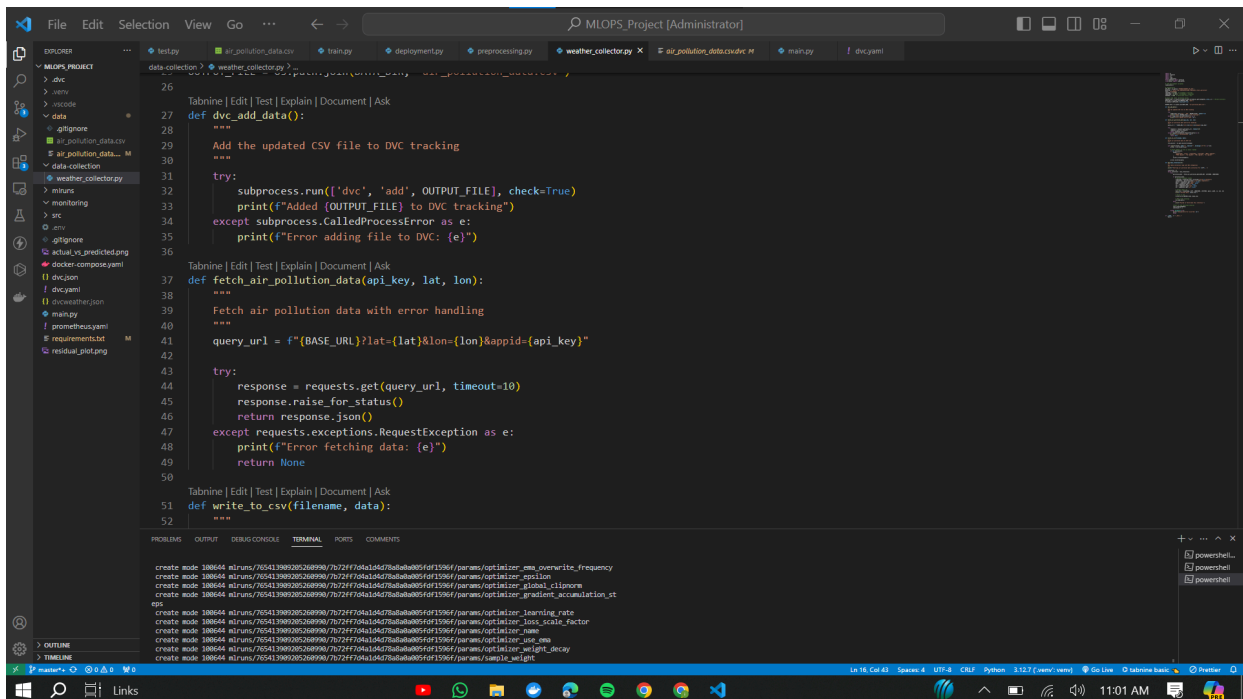
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/optimizer_ema_momentum
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/optimizer_ema_overwrite_freq
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/optimizer_epsilon
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/optimizer_global_clipnorm
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/optimizer_gradient_accumulat
eps
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/optimizer_learning_rate
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/optimizer_loss_scale_factor
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/optimizer_name
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/optimizer_use_ema
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/optimizer_weight_decay
create mode 100644 mlruns/765413909205260990/7b72ff7d4a1d4d78a8a0a005fdf1596f/params/sample_weight
```

Add a python function named **fetch_weather_data** which fetches the data from the website and stores it in the csv file called “**air_pollution_data.csv**”. The screenshot below shows the sample data from the csv file.

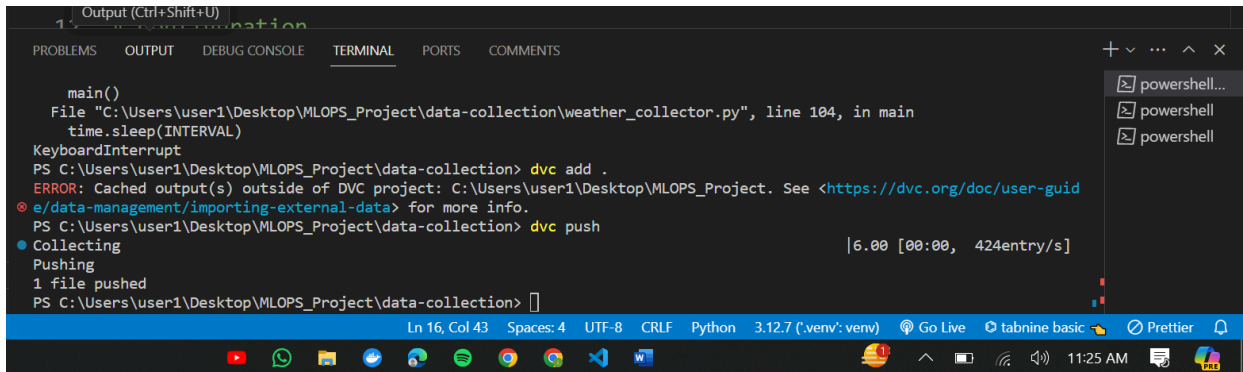


The script code is the following

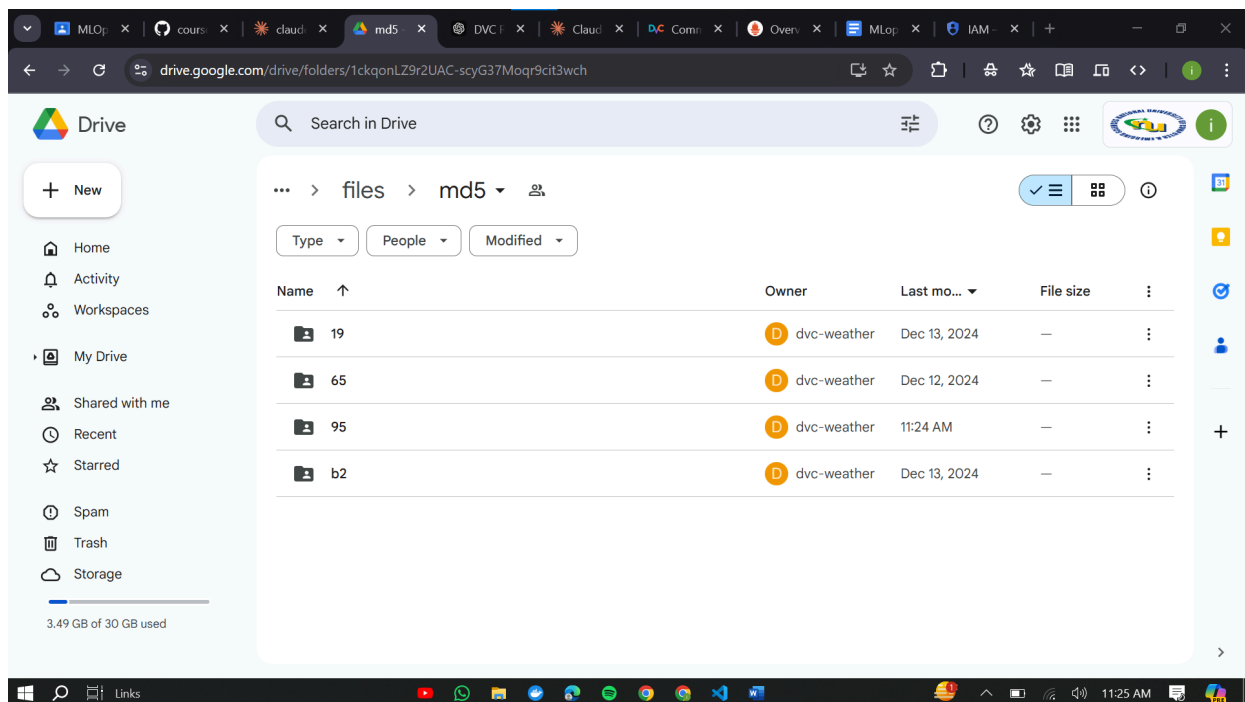


Version Control with dvc:

Everytime when a new data append in the csv file, dvc tracks the changes and with the help of these commands (dvc add, dvc push) the data will be pushed into the remote storage as shown in the screenshot below

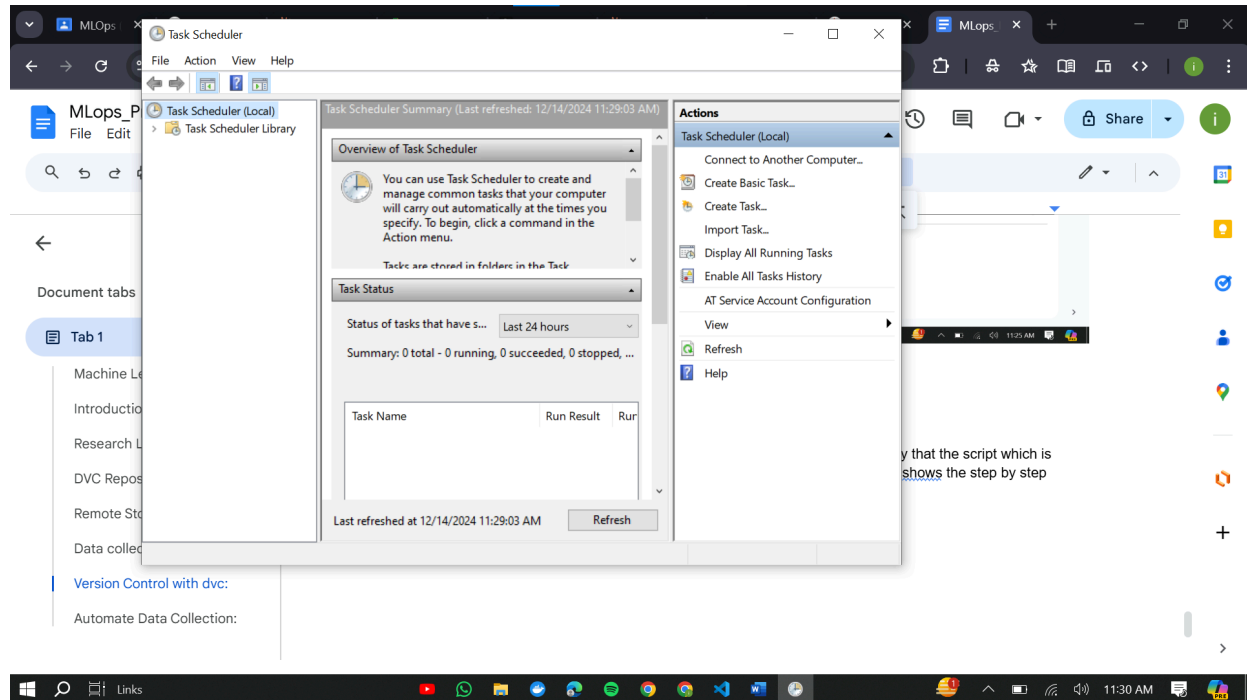


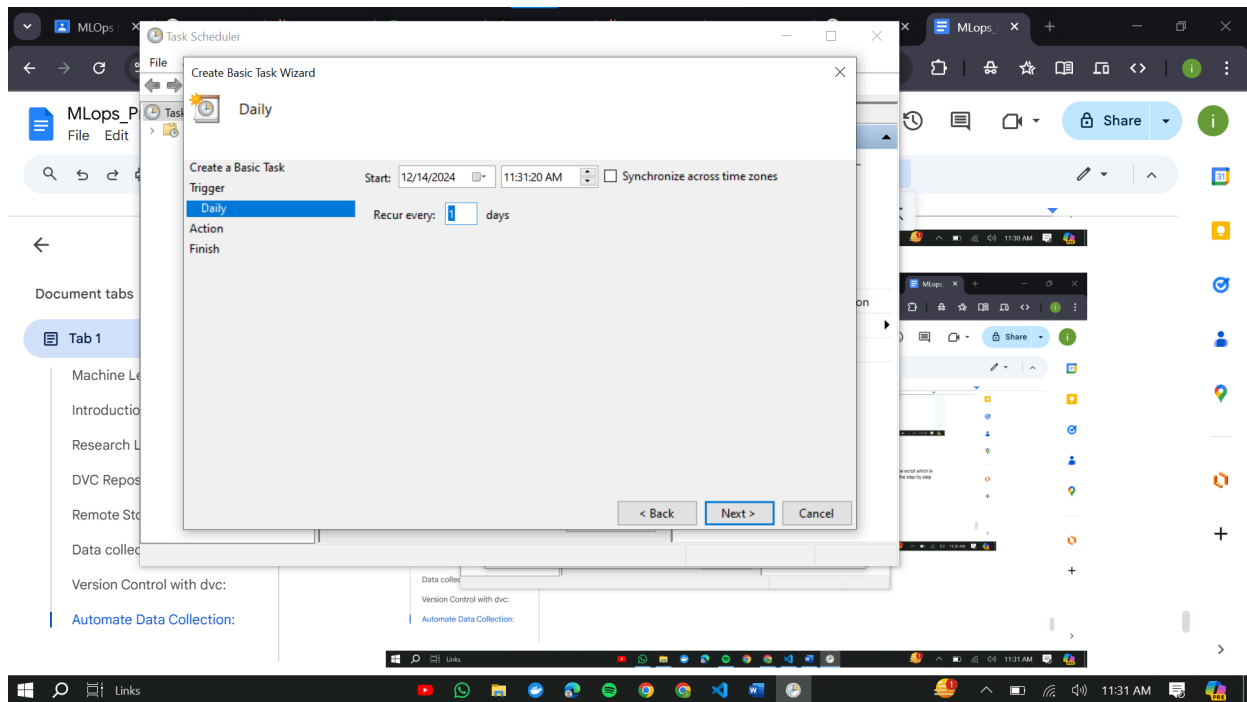
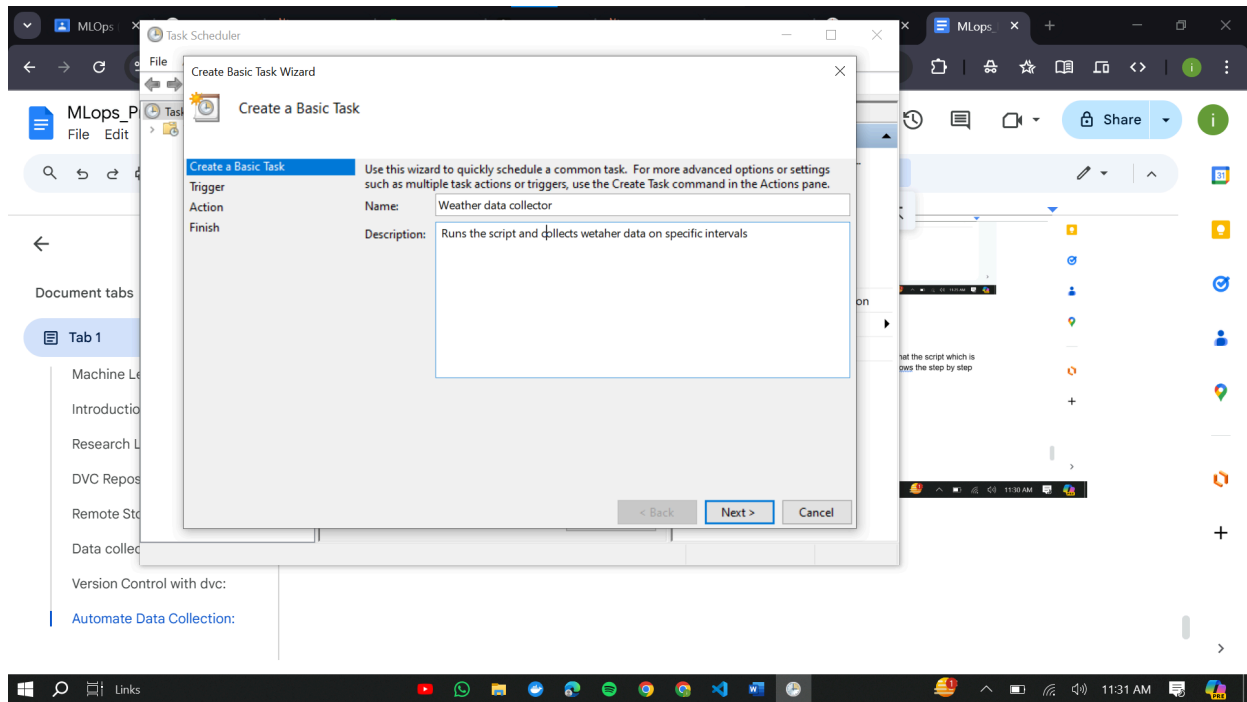
```
main()
File "C:\Users\user1\Desktop\MLOPS_Project\data-collection\weather_collector.py", line 104, in main
time.sleep(INTERVAL)
KeyboardInterrupt
PS C:\Users\user1\Desktop\MLOPS_Project\data-collection> dvc add .
ERROR: Cached output(s) outside of DVC project: C:\Users\user1\Desktop\MLOPS_Project. See <https://dvc.org/doc/user-guide/data-management/importing-external-data> for more info.
PS C:\Users\user1\Desktop\MLOPS_Project\data-collection> dvc push
Collecting [6.00 [00:00, 424entry/s]
Pushing
1 file pushed
PS C:\Users\user1\Desktop\MLOPS_Project\data-collection>
```

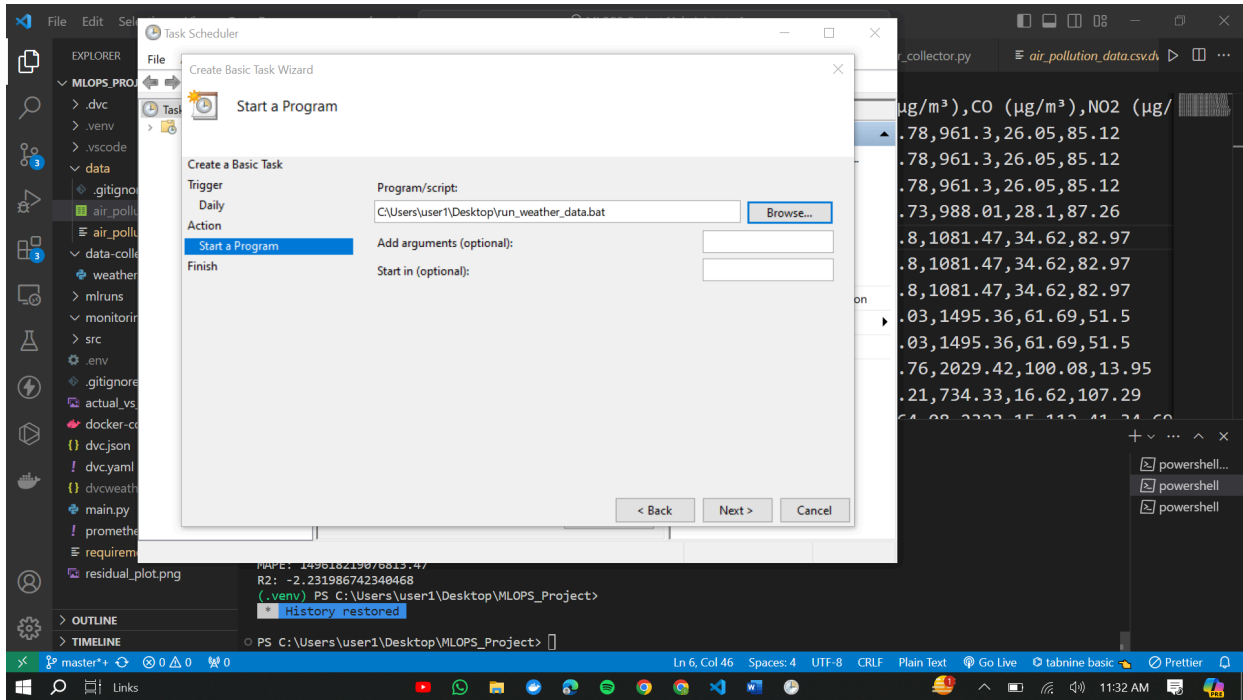


Automate Data Collection:

With the help of task scheduler in Windows, I automated the process in such a way that the script which is responsible of collecting the data is triggered every 24 hrs, the below screenshots shows the step by step process.







Make a windows bat file that automatically triggers the script. The contents of the bat file are

```
@echo off
```

```
Cd C:\Users\user1\Desktop\MLOPS_Project\data-collection\weather_collector.py
```

```
python weather_collector.py
```